Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 20 (2013) 52 – 57

Procedia Computer Science

Complex Adaptive Systems, Publication 3
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2013- Baltimore, MD

Initialization Issues in Self-organizing Maps

Iren Valova[a]*, George Georgiev[b], Natacha Gueorguieva[c], Jacob Olson[a]

[a]University of Massachusetts Dartmouth, 285 Old Westport Rd, N.Dartmouth, MA 02747, USA
[b]University of Wisconsin Oshkosh, 800 Algoma Blvd, Oshkosh, WI 54901,
[c]College of Staten Island, City University of New York, 2800 Victory Blvd, Staten Island, NY 10314, USA

Abstract

In this paper we present analysis and solutions to problems related to initial positioning of neurons in a classic self-organizing map (SOM) neural network. This means that we are not concerned with the multitude of growing variants, where new neurons are placed where needed. For our work, we consider placing the neurons on a Hilbert curve, as SOM have the tendency to converge similarly to self-similar curves. Another point of adjustment in SOM is the initial number of neurons, which depends on the data set. Our investigations show that initializing the neurons on a self-similar curve such as Hilbert provides a quality coverage of the input topology in much less number of epochs as compared to the usual random neuron placement. The meaning of quality is measured by absence of tangles in the network, which is one-dimensional SOM utilizing the traditional Kohonen training algorithm. The tangling of SOM presents the problem of topologically close neighbors that are actually far apart in the neuron chain of the 1D network. This is related to issues of proper clustering and analysis of cluster labels and classification. We also experiment and provide analysis where the number of neurons is concerned.

Selection and peer-review under responsibility of Missouri University of Science and Technology

Keywords: self-organizing maps, Hilbert curves, neurons, SOM chains

1. Introduction

Self-organization is one of the closest artificial neuron architectures to the human brain. In this work, we are presenting the self-organizing map (SOM) neural network as introduced first by Kohonen [1] in the form of a one-dimensional chain. This means that every neuron has at the most two neighbors (one to the left and one to the right). While Kohonen presented a proof that the chain will converge and detangle eventually, the problem is how many epochs, or showings of the data set this will take. Our work presents a possible solution to this problem as well as a detailed analysis of it and concerns typical for the SOM architecture.

A classic SOM may have two typical configurations – one-dimensional or two-dimensional with various

* Corresponding author. Iren Valova, Tel.: 5089998502; fax: 5089999144.
E-mail address: ivalova@umassd.edu.

Selection and peer-review under responsibility of Missouri University of Science and Technology
doi:10.1016/j.procs.2013.09.238

neighborhood formats (hexagonal, square, etc.). In his presentation of SOM, Kohonen theoretically proved that the one-dimensional configuration will converge in a detangled state after a number of epochs, where one epoch constitutes presenting the entire data set to the network in a random order. No such proof has been offered for the two-dimensional SOM configurations. The SOM operates on finding the closest neuron to a data point via a distance function and then moving that neuron, and potentially its neighbors, towards that data point. Thus, SOM creates a map of the input data space, effectively translating a multidimensional input to a cluster map where the inherent groupings/clusters of the data set become obvious. SOM typically handles topological mapping [2, 3], density distribution and clustering problems [4, 5]. It requires some form of cluster analysis tool in order to handle classification tasks [6, 7].

For the sake of comprehensive presentation, it must be mentioned that there are growing architectures of SOM [8, 9, 10] which do not experience the same initialization/tangling problems. However, the main aim of our work is the improvement of the classic one-dimensional SOM as being the one that is most widely used in industry [11, 12].

2. Initialization of SOM

2.1. Classic SOM operation

The SOM algorithm can be briefly described as follows. We are working with a one-dimensional chain of neurons as shown in Fig. 1. Initially, the neurons are placed either randomly around the input space, or start from one point, or be placed along the turns of a Hilbert curve as shown in Fig. 2. The number of neurons must be decided before the network can begin processing the input data. Since SOM are learning without a teaching signal, no apriori knowledge of the data set is assumed. This means that we have no information on how many data points there are or how these data points may be distributed. This, in turn, means, that no information on how many neurons might be optimal is available. This is one of the problems we analyze in our work.

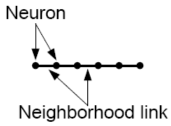

Fig. 1. Example of one-dimensional neighborhood

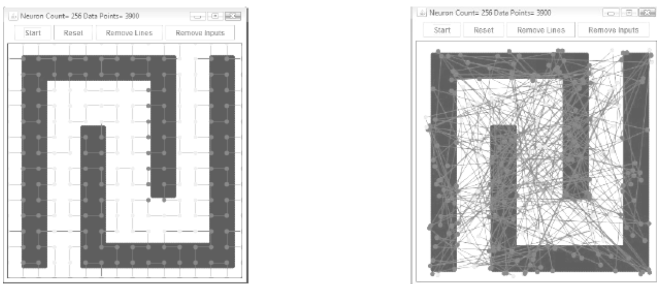

Fig. 2. Hilbert and random initializations of one-dimensional SOM

The SOM algorithm calls for weights, which have the same vector dimensionality as the input data. For the purposes of visualization, we are using two-dimensional toy data sets. Therefore, the weight vectors for the neurons are two-dimensional. The selection of a winning neuron is accomplished by measuring the Euclidean distance between the data point vector and the neuron weight vector. Once a winning neuron is selected, its weight is updated based on: $w_{ij}(new) = w_{ij}(old) + \alpha[x_j - w_{ij}(old)]$, while the neighbors are updated based on:

$$w_{(i-n)j}(new) = w_{(i-n)j}(old) + \alpha[x_i - w_{(i-n)j}(old)] \textit{ to update neighbor before winner}$$
$$w_{(i+n)j}(new) = w_{(i+n)j}(old) + \alpha[x_i - w_{(i+n)j}(old)] \textit{ to update neighbor after winner}$$

The learning rate is also an adjustable parameter as is the neighborhood size. The learning rate can be treated either as a rigid parameter or can be decaying with the epochs.

2.2. Tangling and number of neurons for initialization

The issue of tangling presents a problem when it comes to topological mapping of the input data and classification. Tangling is defined as two neurons converging as topological neighbors but starting as map neighbors. This is best illustrated in Fig. 3. The problem is in the fact that the two neurons which should be far apart have ended up representing the same grouping or cluster of input data. This is diminishing the quality of topological mapping as measured by benchmarks [13], as well as preventing further utilization of SOM as input to classification neural networks. We will be demonstrating both issues in the Results section.

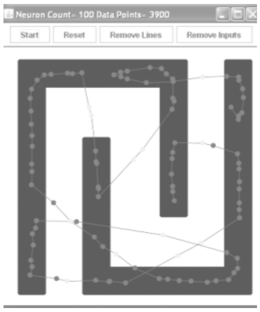

Fig. 3. Tangling in a 100-neuron SOM

Assume, for example, that we eliminate the useless neurons (shown in green in Fig. 3), i.e. the ones not covering the input space (shown in black in Fig. 3). Then, we will have sections of SOMs, which can be reconfigured to represent contiguous input. In such case the tangling becomes a problem, as the reconfiguration will be unmanageable (we cannot assume topological neighbors to be actual map neighbors). Aside from this, topologically speaking, we could liken the problem as calling somebody from the other side of town to come and represent our neighborhood which is far away. Obviously, this will skew the statistical representation/coverage of the neighborhood.

3. Results and discussion

3.1. Hilbert curves

The main reason for utilizing this particular type of neuron order is due to the fact that Hilbert curves represent the type of space filling curves towards which SOM naturally converges [1]. Since recursion is a phenomenon observed in nature and space filling curves are very similar to fractals, utilizing Hilbert is an intuitive choice. Space filling curves, unlike fractals, however, have integer dimension and are called that because the limit curve defined by them is one that fills higher dimension space.

Another aspect of this choice is the ability to increase and decrease the number of neurons simply by increasing or decreasing the order or iteration of the curve as shown in Fig. 4. The Hilbert curve represents the limit curve reached when a space is subdivided into infinite number of subdivisions. For the purposes of our experiments we have introduced two-dimensional Hilbert curves with a recursive production rule.

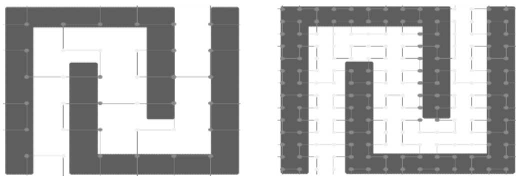

Fig. 4 Hilbert curves accommodating 50 and 100 neurons respectively

The neuron colors in Fig. 4 do not have significance at this stage of the network operation. When the network has converged, the green neurons are the ones that do not cover input space and could be considered redundant.

Lastly, we should mention that with Hilbert curves being space filling, we must establish the boundaries of the input space ahead of time. We basically construct a rectangular window taking into account the furthest points from the data set.

3.2. Experimental results

For each experiment we present a table outlining the parameters of the utilized SOM. This is followed by random initialization and its result after 200 epochs, followed by Hilbert initialization and its result after 200 epochs.

Number of Inputs	3900
Num Neurons (Random)	50
Num Neurons (Hilbert)	64
Alpha	0.6
Alpha Decay Value	0.001
Density Modification	4
Neighborhood Range	2

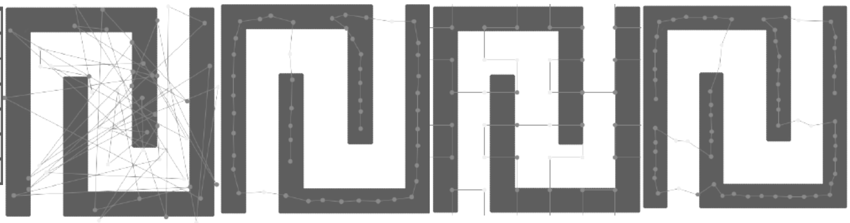

Fig. 5 SOM with 50 neurons. Both random and Hilbert initialization do not show tangling

Fig. 5 demonstrates the mapping outcome for a SOM with 50 neurons (64 in the case of Hilbert as necessitated by the nature of the curve). While both initialization methods do not show any tangling, the coverage of the input space (two interlocked spirals featuring 3900 input points) is sparse and can be considered as an outline at best. This demonstrates that with a small and insufficient number of neurons the network will perform well, but can hardly be considered useful.

Fig. 6 demonstrates the opposite – 200 neurons leading to extreme tangling in the case of random initialization. The Hilbert curve leads to a considerably better result with minimal tangling. It should be noted that an initial neighborhood size of zero, i.e. only the winning neuron updating its weights, the neurons are evenly distributed through the input space, but the tangling is extreme with random initialization. The Hilbert initialization, while featuring very few tangles, illustrates the large number of superfluous neurons. With this experiment, we show not only the tangling perspective of initialization, but also the need to utilize the optimum number of neurons for the data set. We should mention that this is the next step of our research – optimization of the classic SOM parameters without introducing bottlenecks in complexity and running time of the algorithm.

Finally, Fig. 7 demonstrates a SOM with 100 neurons, decaying learning rate and neighborhood range of 2 as specified in the table of Fig. 7. No tangling is evident in the Hilbert initialized SOM, while the randomly initialized network does show topological closeness for neurons that are not physical neighbors. The input space has sufficient coverage for the random and very nicely distributed solid coverage with Hilbert initialization.

Number of Inputs	3900
Num Neurons (Random)	200
Num Neurons (Hilbert)	256
Alpha	0.6
Alpha Decay Value	0.001
Density Modification	4
Neighborhood Range	0

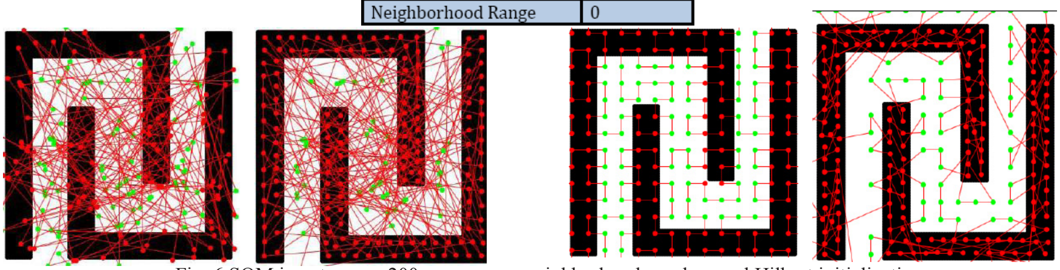

Fig. 6 SOM in extreme – 200 neurons, no neighborhood, random and Hilbert initialization

Number of Inputs	3900
Num Neurons (Random)	100
Num Neurons (Hilbert)	256
Alpha	0.6
Alpha Decay Value	0.001
Density Modification	4
Neighborhood Range	2

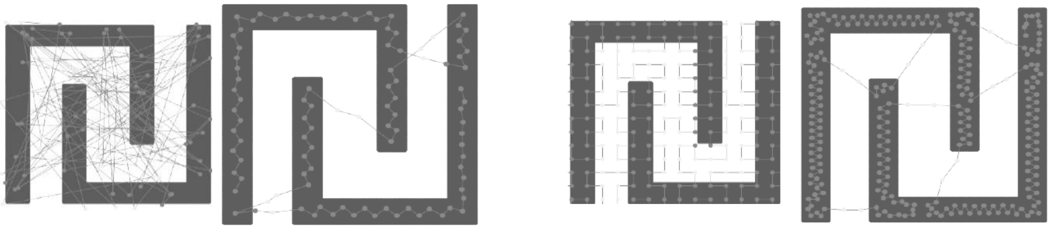

Fig. 7 SOM with 100 neurons, random initialization has tangling, while Hilbert provides nicely distributed coverage with no tangles

Number of Inputs	3900
Num Neurons (Random)	200
Num Neurons (Hilbert)	256
Alpha	0.6
Alpha Decay Value	0.001
Density Modification	4
Neighborhood Range	4

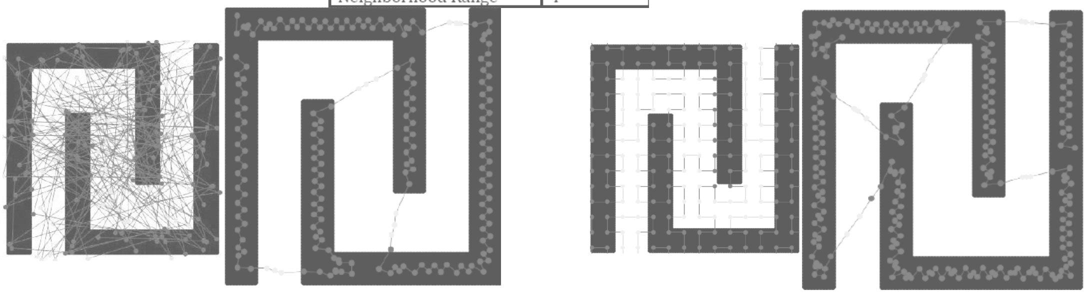

Fig. 8 SOM with 200 neurons, large neighborhood of 4 and decaying learning rate

The neighborhood size influence is evident in Fig. 8 where we demonstrate that with the increase of neighborhood size, the tangling decreases even for random initialization, but the distribution of neurons through the input space become more uneven. The neurons are staying closer together, which is to be expected. However, while it is expected, the parameters' "negotiation" is evident in this experiment – the quality of the coverage is somewhat sacrificed for reduced tangling. Further increasing the neighborhood size from 4 to 6, the result is worsening of the input space coverage, but no tangling at all. Fig. 9 shows only the results of both random and Hilbert initialization.

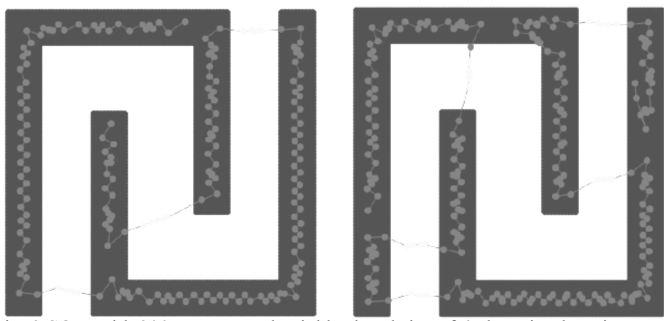

Fig. 9 SOM with 200 neurons and neighborhood size of 6, decaying learning rate

The next experiment features 200 neurons and neighborhood size of 2 where all neurons start from the same one point. With this initialization, the coverage is denser around the starting point and random performs without tangles for the first *n* neurons (Fig.10). For the remainder of the chain, however, the quality of coverage decays for random initialization. Hilbert initialization does not provide the opportunity to start off the neurons from one point.

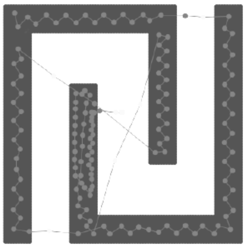

Fig.10 SOM with 200 neurons

Fig. 11 SOM result serving as MLP input and resulting classification of normally distributed data points within the data space box.

3.3. Discussion

So far, in our experiments we have shown that Hilbert initialization performs considerably better than random taking into account neuron distribution across input space, positioning neighboring neurons topologically close on the map and relative independence from neighborhood size and number of neurons in the network.

Before we conclude, one more experiment will be featured. Suppose that the results from SOM were to be used as input to a classifier, e.g. multilayer perceptron (MLP). The SOM chain is broken down for the two separate clusters it covers (each of the spirals) and the MLP is trained on the chains. Further, a testing space of random points within the whole square is fed into the MLP, which is able to classify the space into two separate regions as shown in Fig.11. The SOM features Hilbert initialization with 256 neurons, decaying learning rate and neighborhood size of 2. The network is run for 200 epochs before breaking the chains and feeding the neuron coordinates as training data to the three-layer MLP. The purpose of this experiment is to demonstrate the need for an evenly distributed quality coverage with sufficient number of neurons and the need to have as few tangles in the resulting SOM as possible to ensure the proper analysis of the topological map of the input space.

4. Conclusion

In our experiments we intentionally utilized a two-dimensional toy data set to visualize the issues of coverage, tangling and quality of resulting maps when SOM converges. Given the experiments, the neighborhood size as well as the mode of initialization proved to be the most important when it comes to quality coverage with evenly distributed neurons. We have clarified why tangling is an issue and we have demonstrated that Hilbert curve initialization is ultimately the best tool for the initial placement of neurons. The optimization of the neighborhood size and the learning rate decay are points of future research for us.

References

1. T.Kohonen, Self-organizing maps, 2nd edition, Springer, New York, 1995.
2. J.Jockusch, H.Ritter, An instantaneuos topological mapping model for correlated stimuli, Proceedings of IJCNN, p 529-34, 1999.
3. E. Arsuaga Uriarte, F. Díaz Martín, Topology preservation in SOM, International Journal of Applied Mathematics and Computer Sciences, p.19-22, 2005.
4. J.Vesanto, Clustering of the self-organizing map, IEEE Transactions on Neural Networks, 11(3) 586-600, 2000.
5. S.Gunter, H.Bunke, Self-organizing map for clustering in the graph domain, Pattern Recognition Letters, 23(4) 405-417, 2002.
6. H.Kauppinen, H.Rautio, O.Silven, Non-segmenting defect detection and SOM-based classification for surface inspection using color vision, Proc. SPIE 38826, 270, 1999.
7. A.Ultsch, F. Mörchen. ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM. Univ. 2005.
8. A.Rauber, D.Merkl, M.Dittenbach, The growing hierarchical self-organzing map: exploratory analysis of high-deminsional data, IEEE Transactions on Neural Networks, 13(6) 1331-1341, 2002.
9. D.Maclean, I.Valova, Parallel growing SOM monitored by genetic algorithm, Proceedings of IJCNN, p 1697-1702, 2007.
10. B.Fritzke, Growing cell structures – a self-organizing network for supervised and unsupervised learning, Neural Networks, 7(9) 1441-1460.
11. T.Kuremoto, T.Komoto, K.Kobayashi, M.Obayashi, Parameterless growing SOM and its applications to a voice instruction learning system, Journal of Robotics, Vol.2010, Article ID 307293, doi:10.1155/2010/307293, 2010
12. D.Brown, I.Craw, J.Lewthwaite. A SOM based approach to skin detection with application in real time systems, Proc. of the British Machine Vision Conference, vol. 2, p 491-500. 2001
13. D.Beaton, I.Valova, D.MacLean, CQoCO: A measure for comparative quality of coverage and organization for self-organizing maps. Neurocomputing 73.10 (2010): 2147-2159.